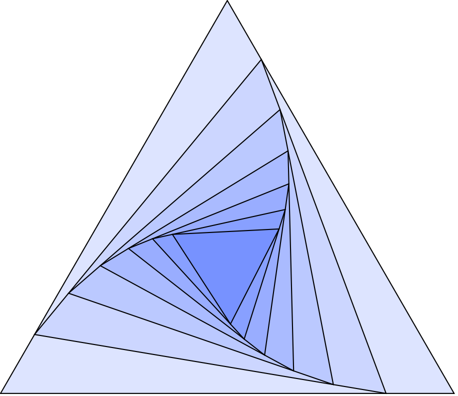


orgmode examples

draw, code evaluation and present in orgmode with LaTeX beamer

kimim

Document ID: Revision: Updated on: September 2, 2021

Outline

Outline

Introduction

Purpose

To evaluate many features of orgmode, such as

- drawing with code
- evaluating results of code snippets
- exporting orgmode file to pdf slides

Introduction

How

Following tools are used in this file:

- MSYS2 provides many tools and libraries
- GraalVM JDK supports Java, JS, R and more
- GNU Emacs with kimim-emacs configuration
- Org Mode, including org-babel, org-export
- TexLive with beamertheme-kimim style
- PlantUML, Graphviz, LaTeX tikz package
- Inkscape to convert svg to pdf, during orgmode-pdf exporting

Outline

Preparation

Emacs settings

You may need to use kimim-emacs configuration:

```
# backup existing emacs config
cd ~ && mv .emacs .emacs-backup && mv .emacs.d .emacs.d-backup
# clone this config
git clone https://github.com/kimim/kimim-emacs
# copy default .emacs to ~
cp kimim-emacs/.emacs ~
```

Preparation

Emacs and Orgmode version

Firstly, let's check GNU Emacs[1] and Orgmode[2] version:

```
(concat  (emacs-version)
         "\nOrgmode " (org-version))
```

```
GNU Emacs 28.0.50 (build 6, x86_64-w64-mingw32)
 of 2021-08-31
Orgmode 9.4.4
```

[1] https://www.gnu.org/software/emacs

[2] https://orgmode.org

Preparation

TexLive and Beamer Theme

Install TexLive[3] to `<texlive-path>` and clone beamertheme-kimim[4], and update TeX cache:

```
git clone https://github.com/kimim/beamertheme-kimim \
    <texlive-path>/texmf-local/tex/latex/beamertheme-kimim
mktexlsr
```

[3] http://tug.org/texlive

[4] https://github.com/kimim/beamertheme-kimim

Preparation

Inkscape version

Install Inkscape[5] to convert SVG image to PDF.
This is inkscape version on my Windows 10:

```
inkscape --version
```

```
Inkscape 1.0.2-2 (e86c870879, 2021-01-15)
```

[5] https://inkscape.org

Outline

PlantUML

PlantUML settings in Emacs

Download plantuml.jar[6], and set jar-path:

```
(require 'url-handlers)
(require 'ob-plantuml)
(url-copy-file "https://nchc.dl.sourceforge.net/project/plantuml/plantuml.jar"
               "./plantuml.jar" t)
(setq org-plantuml-jar-path "./plantuml.jar")
```

[6] https://plantuml.com

PlantUML

PlantUML version

Here is the version info on my machine, including JVM, dot and graphviz:

```
(shell-command-to-string
 (concat
  "java -jar " org-plantuml-jar-path " -version"))
```

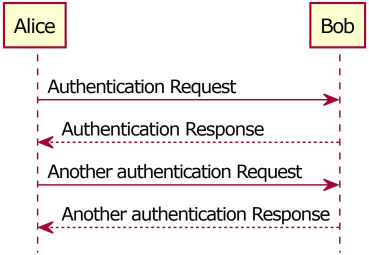
```
PlantUML version 1.2021.8 (Sat Jun 26 16:20:59 CST 2021)
(GPL source distribution)
Java Runtime: OpenJDK Runtime Environment
JVM: OpenJDK 64-Bit Server VM
Default Encoding: Cp1252
Language: en
Country: US

PLANTUML_LIMIT_SIZE: 4096

Dot version: dot - graphviz version 2.44.1 (20200629.0846)
Installation seems OK. File generation OK
```

PlantUML

Sequence Diagram

Let's draw a simple sequence diagram with this plantuml code:

```
@startuml
hide footbox
hide unlinked
Alice -> Bob: Authentication Request
Bob --> Alice: Authentication Response
Alice -> Bob: Another authentication Request
Alice <-- Bob: Another authentication Response
@enduml
```

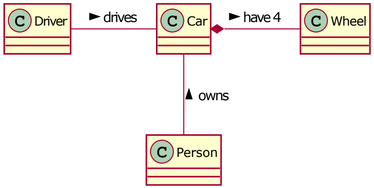

Alice
Bob
Authentication Request
Authentication Response
Another authentication Request
Another authentication Response

PlantUML

Class Diagram

A simple class diagram

```
@startuml
class Car
Driver - Car : drives >
Car *- Wheel : have 4 >
Car -- Person : < owns
@enduml
```

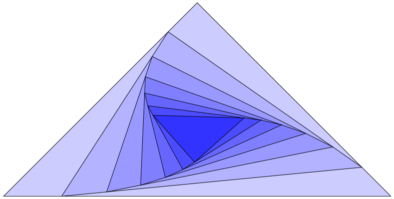


Outline

Tikz

tikz diagram

```
\begin{tikzpicture}
  \coordinate (A) at (0,0);
  \coordinate (B) at (60, 0);
  \coordinate (C) at (30, 30);
  \foreach \density in {20,30,...,80}{%
    \draw[fill=blue!\density]
    (A)--(B)--(C)--cycle;
    \path
    (A) coordinate (X)
    -- (B) coordinate[pos=.15](A)
    -- (C) coordinate[pos=.15](B)
    -- (X) coordinate[pos=.15](C);
  }
\end{tikzpicture}
```



Outline

Org-babel Evaluating Programming Languages

emacs lisp

```
(emacs-version)
```

```
GNU Emacs 28.0.50 (build 6, x86_64-w64-mingw32)
 of 2021-08-31
```

Org-babel Evaluating Programming Languages

shell

```
sh --version
```

```
GNU bash, version 5.1.8(1)-release (x86_64-pc-msys)
Copyright (C) 2020 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>

This is free software; you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
```

Org-babel Evaluating Programming Languages

C

```
printf("%s is %d years old\n", "C programming language", 2021-1972);
```

```
C programming language is 49 years old
```

Org-babel Evaluating Programming Languages

C++

```
cout << "C++ is " << 2021-1979 << " years old" << endl;
```

```
C++ is 42 years old
```

Org-babel Evaluating Programming Languages

Clojure

Org-babel Evaluating Programming Languages

ClojureScript

Org-babel Evaluating Programming Languages

Java

Org-babel Evaluating Programming Languages

Python

```
python --version
```

```
Python 3.9.6
```

```python
print("Python in Emacs/orgmode")
```

```
Python in Emacs/orgmode
```

Org-babel Evaluating Programming Languages

Rust

```
cargo install rust-script
```

```
(package-install 'ob-rust)
```

```
fn main() {
    println!("Rust in Emacs/orgmode");
}
```

Org-babel Evaluating Programming Languages

Go

```go
package main
import ("fmt")

func main(){
   fmt.Println("emacs")
}
```

Org-babel Evaluating Programming Languages

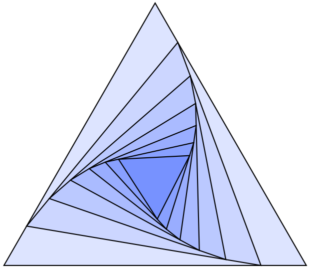
R

Outline

Conclusion

Key Takeaways

- Emacs is a long lasting, and wonderful text editor
- Orgmode is an awesome plain text format
- LaTeX and Beamer is great typesetting tool
- Thus, drawing plantuml diagram with these tools is cool!



Appendix

References I